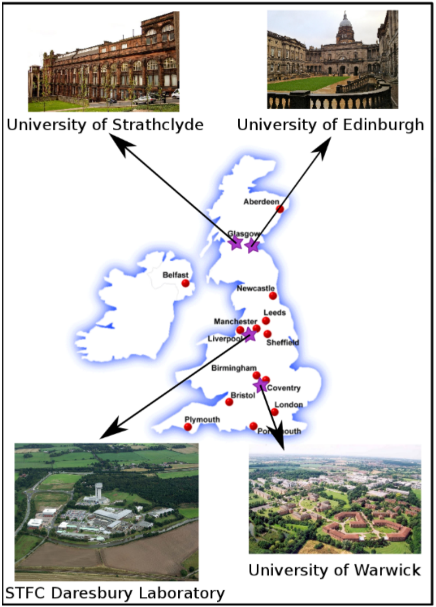
An Evaluation of Popular Noise Reduction Techniques for Multi-scale Problems Involving Particle Flow Simulations

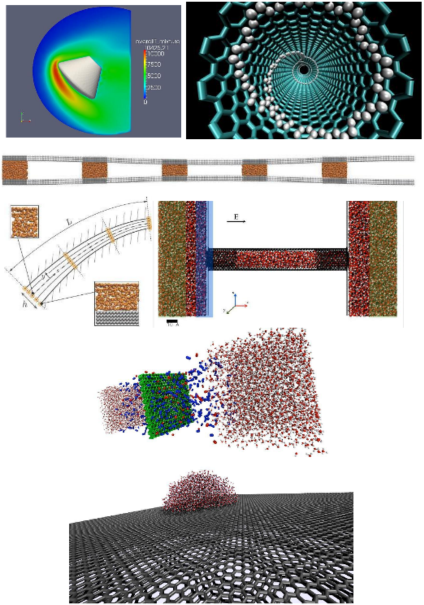
Małgorzata J Zimoń[1,2]
Leopold Grinberg[3], Jason M Reese[4], David R Emerson[2]

[1] James Weir Fluids Lab, University of Strathclyde, Glasgow G1 1XJ, UK;
[2] STFC Daresbury Laboratory, Warrington WA4 4AD, UK;
[3] IBM T.J. Watson Research Center, 1 Rogers St, Cambridge MA 02142, USA;
[4] School of Engineering, University of Edinburgh, Edinburgh EH9 3JL, UK.

malgorzata.zimon@strath.ac.uk

November 26, 2014


University of Strathclyde
University of Edinburgh
Aberdeen
Glasgow
Belfast
Newcastle
Leeds
Manchester
Liverpool
Sheffield
Birmingham
Coventry
Bristol
London
Plymouth
University of Warwick
STFC Daresbury Laboratory



Motivation

- The need for suppressing noise in particle-based simulations in the process of obtaining an ensemble solution.

⇒ Extracting the *true* information, providing *clean* particle distribution functions and smooth gradients from data with low signal-to-noise ratio (SNR).

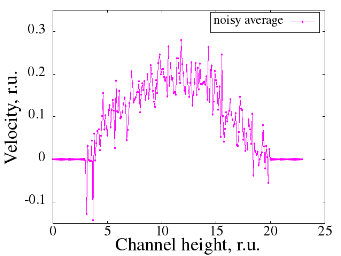


Fig: Noisy velocity profile obtained with molecular dynamics.

Objective

- Development of filtering techniques that extract significant structures from noisy data.
- Investigation of the capabilities of a number of methods, including proper orthogonal decomposition (POD), singular spectrum analysis (SSA), and wavelet thresholding, to overcome unwanted fluctuations in analysed results.

Molecular dynamics (MD)

MD- particle-based modelling technique for solving many-body problems from various fields

- Based on classical mechanics- solving numerically Newton's equation of motion for the interacting (through e.g. pair potentials) multi-particle system.
- The Lennard-Jones potential is often used to model van der Waals (short range) interactions.
- Modelling software- OpenFOAM[1].

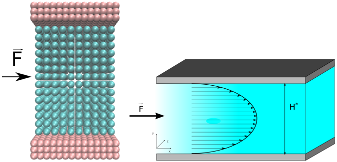


Fig: Poiseuille flow simulated with MD.

Dissipative particle dynamics (DPD)

DPD- an off-lattice, discrete particle modelling method for mesoscopic systems

- Dissipative particle- a centre of mass of mesoscopic portion of fluid (e.g. 3 water molecules in one bead).
- The equations governing time evolution resemble those of MD.
- $\vec{F}_i$ is a sum of pairwise forces (C-conservative, D-drag and R-random): $\vec{F}_i = \sum_{j\neq i}^{N}(\vec{F}_{ij}^C + \vec{F}_{ij}^D + \vec{F}_{ij}^R)$.
- Classic DPD: $\vec{F}_{ij}^C = A_{ij}\left(1 - \frac{r_{ij}}{r_c}\right)\frac{\vec{r}_{ij}}{r_{ij}}$.
- Simulations performed with DL_MESO[2].

1 *www.OpenFOAM.org*

2 *www.stfc.ac.uk/SCD/research/app/ccg/software/DL_MESO/40694.aspx*

Direct simulation Monte Carlo (DSMC)

DSMC- dominant method for performing numerical simulations of rarefied gases when the *Knudsen number* (Kn) is significant. Knudsen number is defined as the ratio of the molecular mean free path and the geometric characteristic length of the flow.

- Uses Monte Carlo simulation to solve the Boltzmann equation.
- One of the main difficulties is reducing the statistical scatter in the results, particularly in low velocity applications.
- Modelling software- dsmcFOAM in OpenFOAM.

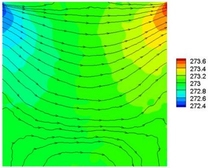


Fig: Heat flux stream traces overlaid on temperature contours computed by DSMC for the driven cavity (at $Kn = 0.2$ and $uw = 10\frac{m}{s}$)[3].

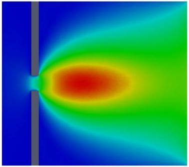


Fig: DSMC simulation of supersonic free jet expansion into vacuum.

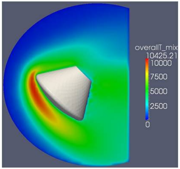


Fig: Temperature (K) contours around a re-entry capsule obtained with dsmcFOAM.

[3] B. John, X.-J. Gu, and D. R. Emerson, "Nonequilibrium gaseous heat transfer in pressure-driven plane poiseuille flow," *Physical Review E*, vol. 88, no. 1, p. 013018, 2013

- POD- a statistical low-rank approximation method that extracts linearly dependent features of data; known also as Karhunen-Loéve decomposition or principal component analysis (PCA).
- POD approximates the *true* data matrix A_{true} from noisy measurements A of rank r; yields matrix $A_k \approx A_{true}$ of rank k ($k < r$) with the lowest error in L_2 or Frobenius norm.
- POD determines a set of orthogonal basis functions (modes, or EOFs[4]) of space and time.
- It can be performed with singular value decomposition (SVD) or eigenvalue decomposition (EVD); SVD can be calculated by means of solving two eigenvalue problems:
$$AA^T = U\Sigma V^T V\Sigma U^T = U\Sigma^2 U^T, \quad A^T A = V\Sigma U^T U\Sigma V^T = V\Sigma^2 V^T.$$

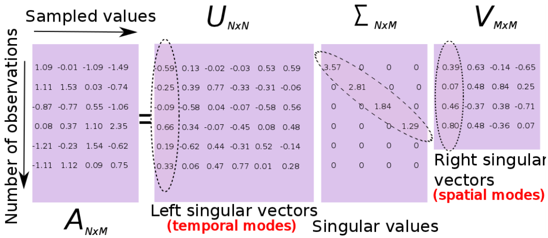


Fig: SVD of matrix *A*.

[4] EOFs- empirical orthogonal functions.

De-noising Through Rank Reduction

- The number of nonzero singular values = rank of the original matrix.
- Calculating the low-rank approximation A_k from matrix $A \equiv$ extracting noise from data.
- New development of the technique based on time windows (WPOD) by Grinberg[5].

How to determine the number of dominant modes?

- Energy content of eigenvalues (λ_k, squares of singular values), or rate of decay.
- Investigation of smoothness of the temporal modes.
- Calculation of Frobenius norm (or L_2) of $A_k = U\Sigma_k V^T$ versus A_{true}.
- Utilising optimal hard threshold (ongoing work).

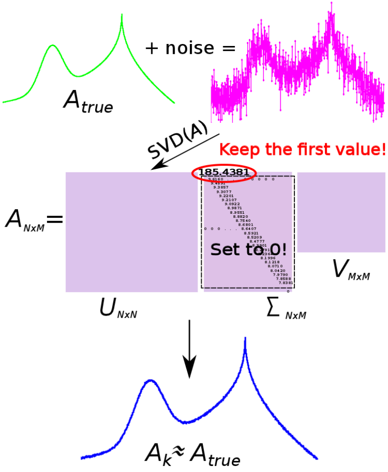
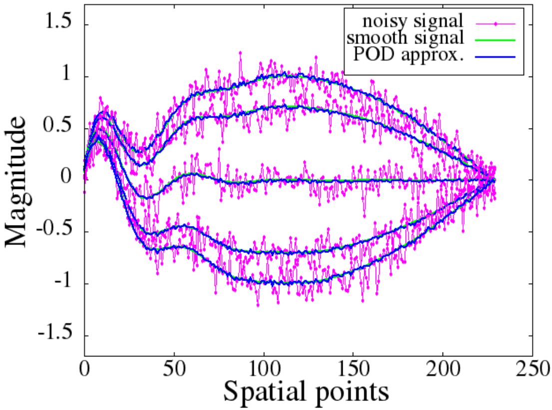
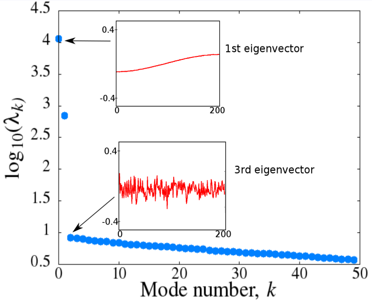


Fig: Reconstruction of the signal with only one dominant mode.

[5] L. Grinberg, "Proper orthogonal decomposition of atomistic flow simulations," *J. Comp. Phys.*, vol. 231, no. 16, pp. 5542–5556, 2012

Analysis of Synthetic Signals





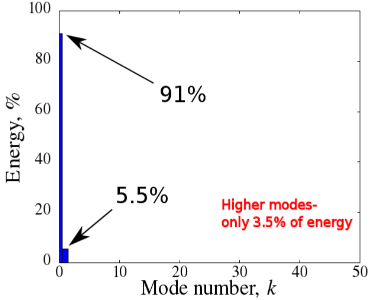


Fig: POD reconstruction of synthetic signal corrupted with Gaussian noise 0.10*randn(); *Right*- rate of convergence and energy of the eigenvalues with corresponding eigenvectors.

WPOD

Moving window: $T_{POD} = N_{POD} N_{ts} \Delta t$, where N_{POD} is the number of time averages used, N_{ts} defines how many observations are in one average and Δt is the simulation time-step.

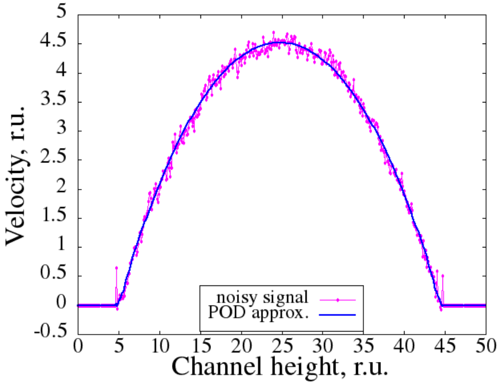
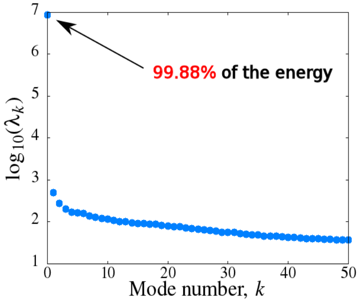
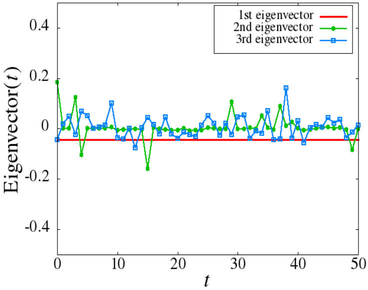


Fig: WPOD analysis of a velocity profile for liquid argon Poiseuille simulation; $N_{ts} = 1$, $N_{POD} = 500$.





Results of Applying WPOD to Particle Data

Steady-state simulation

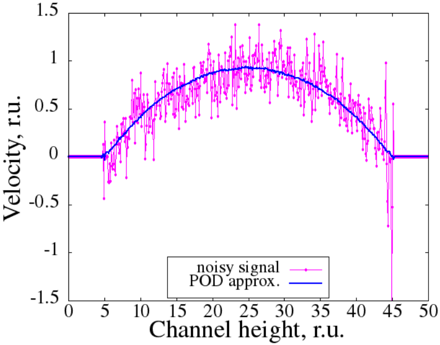


Fig: WPOD reconstruction of a velocity profile for force-driven water flow MD simulation; $N_{ts} = 1$, $N_{POD} = 1000$.

Non-stationary simulation

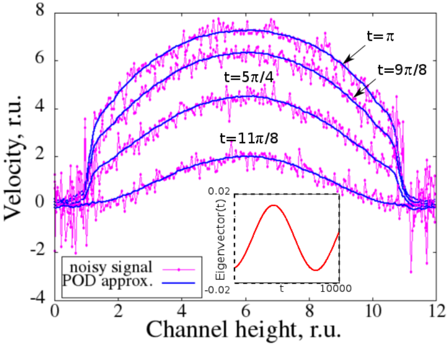


Fig: WPOD applied to the velocity field from periodically-pulsating DPD flow; $N_{ts} = 1$, $N_{POD} = 10000$.

WPOD reconstructions of mass flow rate from oscillating MD flow in axially-periodic converging/diverging channel[6]

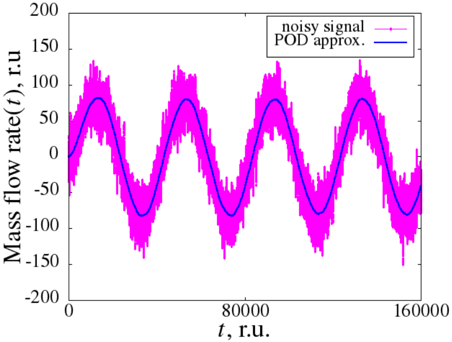


Fig: Constant period of oscillations; $N_{ts} = 1$, $N_{POD} = 160000$.

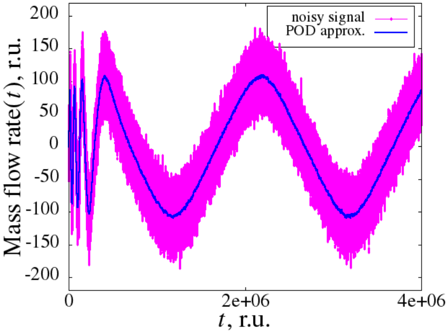


Fig: Mixed-period forcing; $N_{ts} = 1$, $N_{POD} = 4000000$.

[6] M. K. Borg, D. A. Lockerby, and J. M. Reese, "A multiscale method for micro/nano flows of high aspect ratio," *J. Comp. Phys.*, vol. 233, pp. 400–413, 2013

De-noising of velocity from oscillating DSMC gas flow with different number of simulator particles in the domain

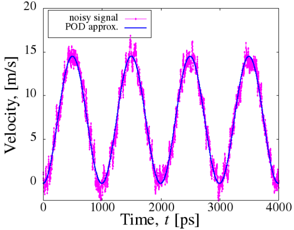


Fig: DSMC simulation with large number of particles: 3,783,495 particles; 50 bins; $N_{ts} = 1$, $N_{POD} = 4000$.

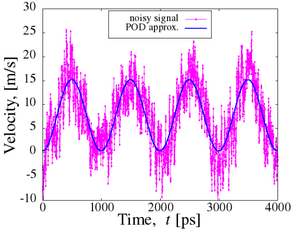


Fig: Medium size system: 189,170 particles, SNR=6.8713 dB; $N_{ts} = 1$, $N_{POD} = 4000$. WPOD achieved SNR=27.6165 dB.

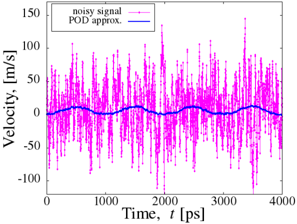


Fig: Small number of particles: only 2,104 particles, SNR=-12.6615 dB; $N_{ts} = 1$, $N_{POD} = 4000$. WPOD achieved SNR= 12.8643 dB.

Strengths

+ POD/WPOD has a capability to successfully separate noise from ensemble average in particle simulations; no assumption of the noise or signal is required.
+ SVD provides the most optimal approximation in spectral (L_2), or Frobenius norm.
+ It is a valuable tool for time-dependent measurements and multi-scale simulations.
+ SVD/EVD can be used to determine when the simulation has converged.

Weaknesses

- POD requires relatively large data-sets if SNR is low.
- Classical SVD is computationally heavy.
- It is not beneficial for steady-state simulations.

1. Can we efficiently de-noise stationary data? SSA, rQRd/urQRd, Wavelet thresholding, WienerChop filter, EMD
2. Can we decrease the processing time? Obtain higher SNR?
   - Random projections, QR instead of SVD? rQRd/urQRd
   - Methods with *a priori* basis? Wavelet thresholding, WienerChop filter
   - Improvement of POD's *cleaning* properties- additional de-noising of dominant eigenvectors? WAVinPOD, POD+SSA, POD+EMD, POD+WienerChop

Discrete Wavelet Transform with Filters

Continuous wavelet transform (CWT)

- **CWT**- scans the signal f with a translated (by u) and scaled (with s) mother wavelet ψ,
  $f_w(u,s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi(\frac{t-u}{s}) dt.$
  - Measures the time-frequency variations of spectral components at various resolutions- *mathematical microscope*.
  - Like Fourier transform, wavelet transform (WT) is a 2D representation of 1D signal f- redundancies.

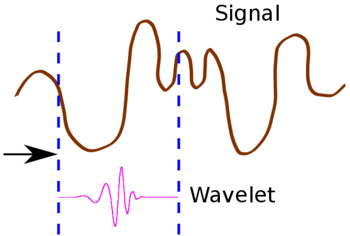


Fig: CWT as a measure of correlation of the signal and wavelet function.

Discrete wavelet transform (DWT)

- **DWT** allows for more practical analysis than endless CWT:
  $$\psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}} \psi\left(\frac{t - k u_0 s_0^j}{s_0^j}\right),$$
  where $s_0 > 1$ is a fixed dilation step, j represents a scale resolution, and $k u_0 s_0^j$ is a discrete shift with a translation factor u_0 .
  - *Dyadic sampling* with $s_0 = 2$ and $u_0 = 1$- orthonormal basis functions.

Multiresolution analysis (MRA)

MRA as a practical design method for orthonormal WT allowing for fast wavelet transform (FWT).

Multiresolution analysis

- In **MRA**, the orthogonal WT decomposes the functional space at a resolution j into a direct sum of orthogonal subspaces of the *large scale* features, V, and *small scale* features, W: $V_j = W_{j-1} \oplus V_{j-1} = W_{j-1} \oplus W_{j-2} \oplus V_{j-2}$.
  - Approximation coefficient c_j and detail coefficient d_j at scale j: projection of $f \subset L^2(\Re)$ onto $V_j \subset L^2(\Re)$ and $W_j \subset L^2(\Re)$, respectively.

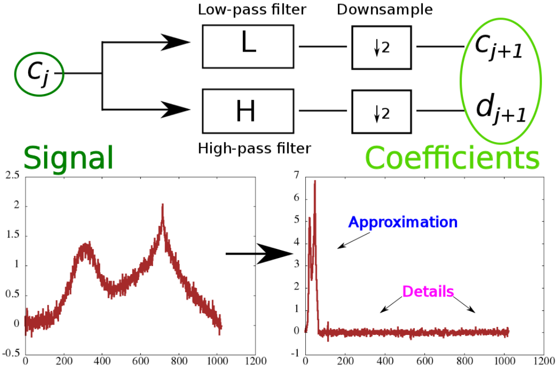


Fig: One step of DWT with filter banks (FWT).

Fast wavelet transform

FWT utilises low- and high-pass quadrature mirror filters in order to compute signal's DWT. Natural extension can be applied to encode two-dimensional data, leading to a set of details in three orientations: horizontal, vertical, and diagonal.

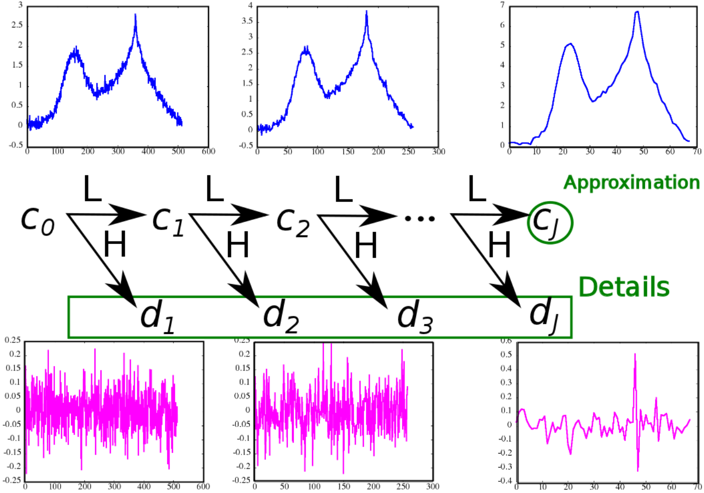


Fig: FWT transforms the signal into a series of details, d_j, and approximation coefficients, c_j.

Universal threshold

The universal threshold (also called *VisuShrink*) is defined as
$T = \sigma_n \sqrt{2 ln(N)}$,
where $\sigma_n = MAD/0.6745$, with MAD being the median absolute value of the finest scale wavelet coefficients, and N the signal length.

Soft and hard thresholding

De-noising by soft thresholding (wavelet shrinkage):

$$\eta_T(d_j) = \begin{cases} sgn(d_j)(|d_j| - T) & if\ |d_j| \geq T, \\ 0 & otherwise. \end{cases}$$

or hard thresholding, defined as follows:

$$\eta_T(d_j) = \begin{cases} d_j & if\ |d_j| \geq T, \\ 0 & otherwise. \end{cases}$$

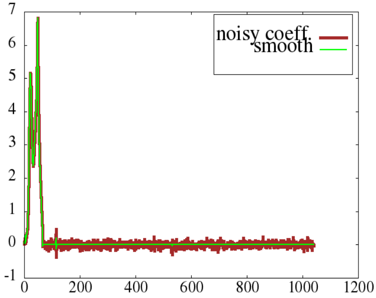


Fig: *Right*- Wavelet coefficients of a disturbed signal and de-noised with soft thresholding.

Weaknesses

- Does not reduce the dimensionality of the data as well as POD!
- For low SNR, the wavelet *follows* the noise.
- Wavelet thresholding is conditioned by a number of parameters.

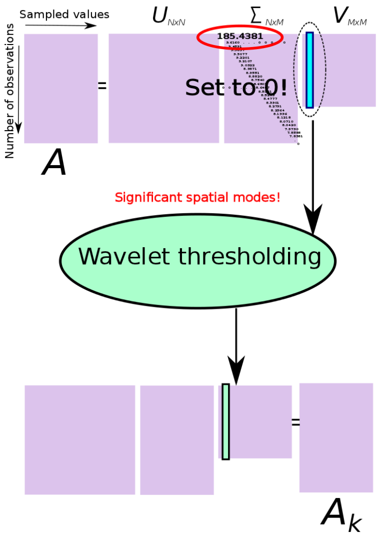


Fig: Schematic of WAVinPOD algorithm.

- *Step 1*: Perform SVD on matrix data A.
- *Step 2*: Define adaptively the number k of dominant modes and set all the higher singular values to zero.
- *Step 3*: Perform WT of the k spatial modes corresponding to the most energetic singular values.
- *Step 3*: Apply wavelet de-noising (soft or hard) with universal threshold to detail coefficients and reconstruct the modes with inverse wavelet transform.
- *Step 5*: Multiply the matrices to construct the low-dimensional approximation of data, A_k.

Strengths

+ More efficient de-noising than POD; higher SNR for smaller number of observations.
+ Preserves SVD's dimensionality reduction; produces approximation with a smaller rank than wavelet thresholding.

Results on Synthetic Data

- Wavelet de-noising based on universal threshold is applied to k-right singular vectors V_k.
- Higher signal-to-noise ratio, computational time much lower than for other wavelets + POD combinations.
- Additional information about the signal is provided in the wavelet domain.

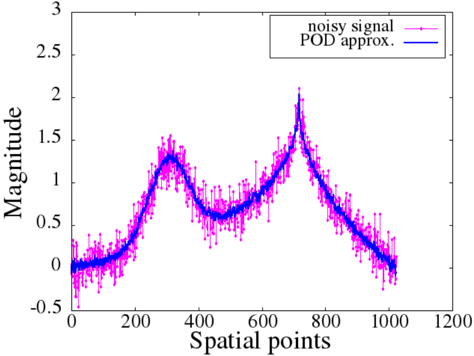


Fig: POD reconstruction of *cuspamax* signal corrupted with noise with SNR=14.8168 dB; 20 observations. Processing time: 0.024953 s. **SNR= 25.1425 dB.**

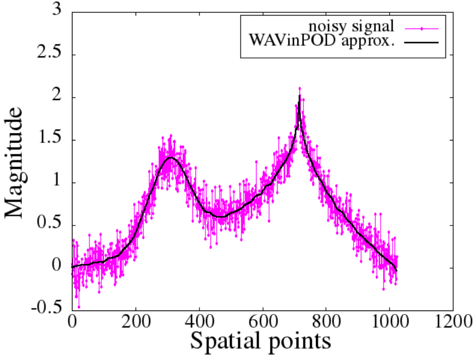


Fig: Wavelet thresholding applied within SVD; 20 observations, filter: db3, 4 levels, hard thresholding. Processing time: 0.036961 s. **SNR= 35.8954 dB.**

Results on Synthetic Data

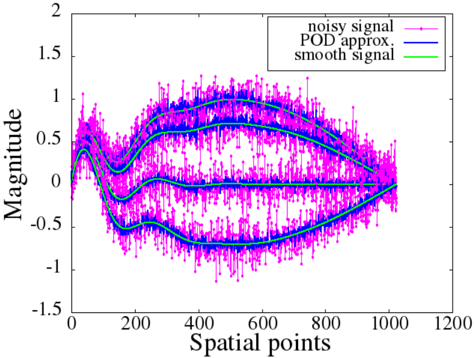


Fig: POD reconstruction of unsteady *fish* signal; 20 observations, 1024 spatial points, SNR of noisy signal=9.2 dB. Processing time: 0.0243 s. **SNR= 19.379 dB.**

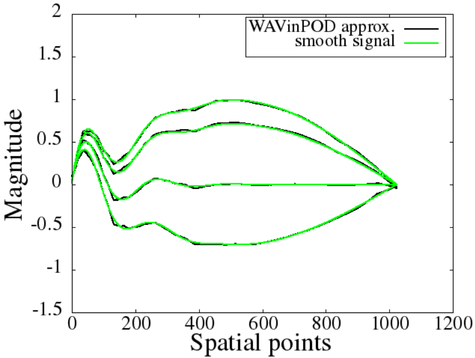
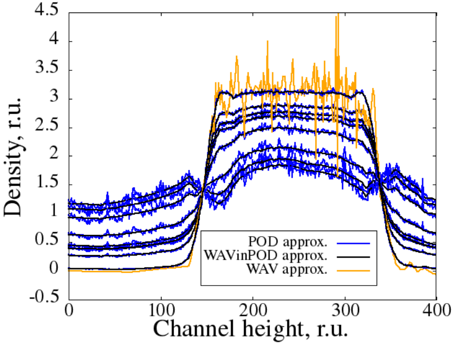
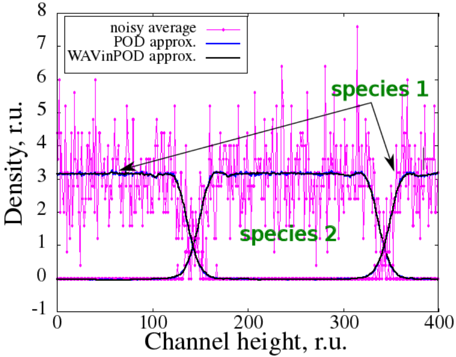


Fig: Thresholding applied within SVD; 20 observations, filter: db3, 6 levels, soft thresholding. Processing time: 0.0389 s[7]. **SNR= 30.3279 dB.**; wavelet thresholding alone provided SNR= 22.2868.

[7] Difference in processing time between POD and WAVinPOD is significantly reduced for larger matrices!

WAVinPOD applied to DPD simulation of phase separation phenomena.





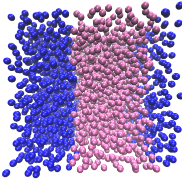


Fig: De-noising with WAVinPOD, wavelet shrinkage (WAV) and WPOD applied to the averaged density field at every $500\Delta t$; $N_{ts} = 10$ and $N_{POD} = 2000$; filter: db3, 6 levels, soft thresholding. *Right*- Comparison of WAVinPOD and WPOD applied to the averaged density field at the last time-step; $N_{ts} = 10$ and $N_{POD} = 2000$. (*Top*). Simulation snapshot (*Bottom*).

The algorithm consists of four main steps : *embedding*, SVD, *eigentriple grouping* and diagonal averaging.

- *Step 1*: In the embedding stage, break the series X of length N into a sequence of *lagged vectors* of size L by forming $X_i = (x_i, \dots x_{i+L-1})^T \quad (1 \leq i \leq K)$. As a result, a *trajectory matrix* H is constructed.
- *Step 2*: Subject the trajectory matrix (Hankel or Toeplitz) to SVD (or EVD).
- *Step 3*: Truncate the singular values.
- *Step 4*: Average over the diagonals yielding a new series $\widetilde{X}$.

Strengths

- + Allows for applying SVD/EVD to one-dimensional data.
- + Unlike WT, does not require *a priori* basis.

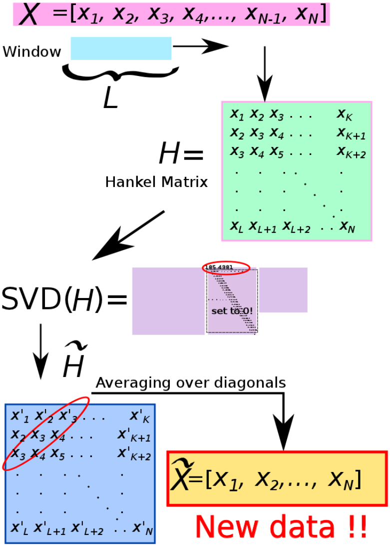


Fig: Schematic of basic SSA.

Multivariate SSA (MSSA)

MSSA A joint trajectory matrix is created from two (or more) data sets $X1$ and $X2$: $Z = (H1; H2)$ or $Z = (H1; H2)^T$, where $H1$ and $H2$ are the matrices formed from the arrays.

Two-dimensional SSA (2D SSA)

2D SSA consist of similar stages as the basic SSA, but utilises 2D window $L_x \times L_y$. The trajectory matrix is a *Hankel block Hankel* matrix of the form:

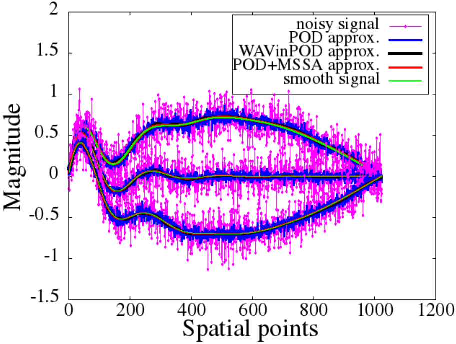
$$H2D = \begin{pmatrix} H_1 & H_2 & H_3 & \dots & H_{K_y} \\ H_2 & H_3 & H_4 & \dots & H_{K_y+1} \\ H_3 & H_4 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ H_{L_y} & H_{L_y+1} & \dots & \dots & H_{N_y} \end{pmatrix}$$

, where every H_j is an $L_x \times K_x$ Hankel matrix built from j-th column of 2D data.

POD+SSA/MSSA !

The algorithm applies **SSA** (or MSSA) to the significant spatial modes obtained with **POD**. Provides improved SNR, and is faster than 2D SSA for large matrices.

De-noising of Synthetic Data



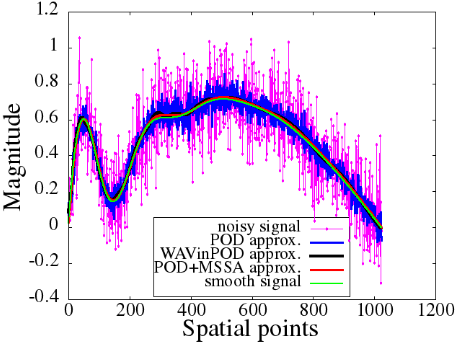


Fig: *Left* and *right*-Comparison of POD, WAVinPOD, and POD+MSSA; 20 observations, filter: db6, 6 levels, hard thresholding, noisy signal **SNR= 9.4413 dB.**

SNR

- POD= **19.7042** dB
- WAVinPOD= **32.3535** dB
- POD+MSSA= 35.3989 dB
  - ⇒ the slowest!

Results of Applying basic SSA to Particle Data

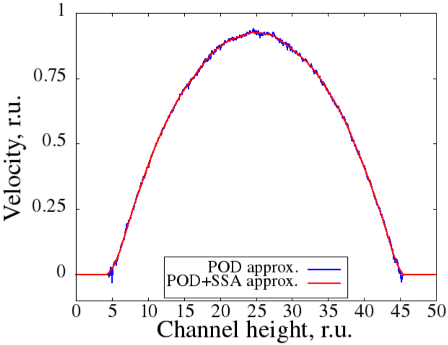


Fig: WPOD compared with SSA in reconstruction of a velocity profile for force-driven water flow MD simulation; $N_{ts} = 1$, $N_{POD} = 1000$.

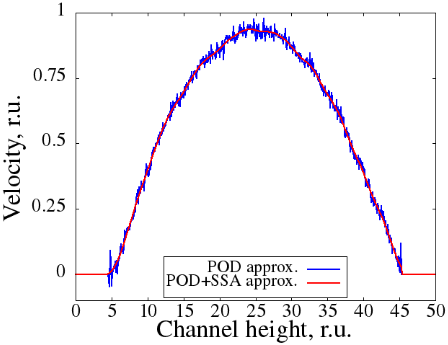


Fig: WPOD compared with SSA in reconstruction of a velocity profile for force-driven water flow MD simulation; $N_{ts} = 1$, $N_{POD} = 100$.

The random QR denoising utilises random projections and QR decomposition instead of truncating the singular values.

- Matrix H_R, containing most of the significant information of the Hankel matrix is obtained by calculating the product of H and a set of P random vectors stored in a matrix Ω: $H_{R(L\times P)} = H_{(L\times K)} \times \Omega_{(K\times P)}$.
- The factorisation of $H_R = QR$ is performed in order to project H onto the reduced rank orthonormal basis Q: $\widetilde{H} = QQ^T H$.
- De-noised series $\widetilde{X}$ is obtained in the same way as in SSA.

uncoiled random QR de-noising (urQRd)

- In **urQRd** the fast matrix-vector multiplication with FFT is implemented to improve the calculation of H_R.

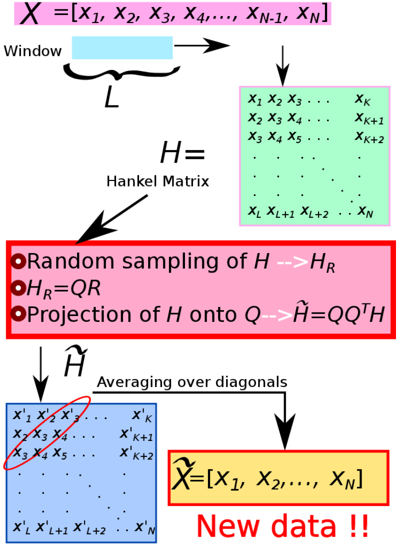


Fig: The rQRd algorithm.

rQRd/urQRd v SSA

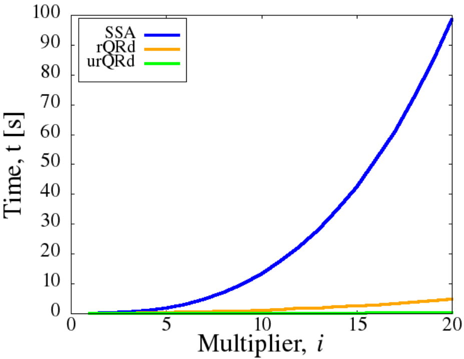


Fig: Comparison of processing time in signal reconstruction with SSA, rQRd, and urQRd. Different lengths of the signal were considered, $N = 1024 * i$, where $i = 1, 2, \ldots, 20$.

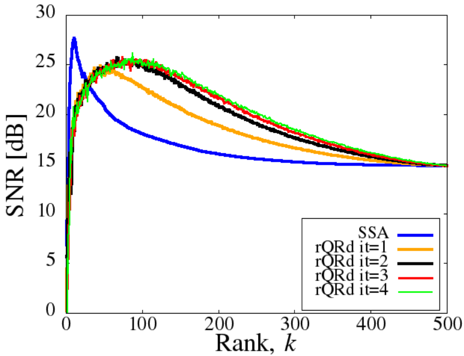


Fig: Values of SNR obtained with SSA and rQRd iterated it=1, it=2, it=3, and it=4 times. The signal considered was Matlab's *cuspamax* having SNR$\approx$ 14.8 dB.

Results obtained with rQRd/urQRd

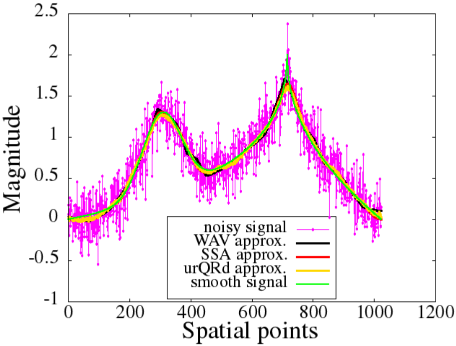


Fig: Result of de-noising Matlab's *cuspamax* signal with wavelet thresholding, SSA, and urQRd; noisy signal SNR= 12.3096 dB, length=1024; filter: db3, 4 levels, hard thresholding.

- Time [s]:
  - SSA ∼0.009, WAV ∼0.007, urQRd ∼0.005;
- SNR [dB]:
  - SSA- 27.17, WAV- 25.55, urQRd- 26.50.

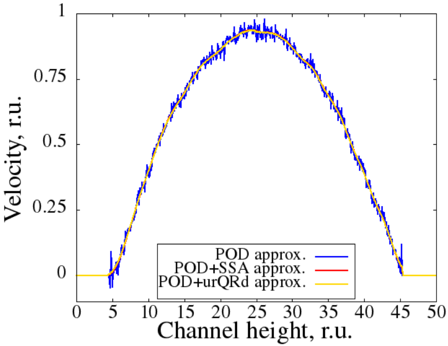


Fig: POD+urQRd produces very close result to POD+SSA reconstruction of a velocity profile for force-driven water flow MD simulation; $N_{ts} = 1$, $N_{POD} = 100$.

EMD

- *Step 1*: Identify the extrema of a given signal $x(t)$.
- *Step 2*: Obtain the envelopes $e_{min}(t)$ and $e_{max}(t)$ by interpolating between minima and maxima, respectively.
- *Step 3*: Compute the mean of the two envelopes, $m_1^1(t) = \frac{e_{min}(t)+e_{max}(t)}{2}$.
- *Step 4*: Extract the detail by subtracting the mean from the signal, $h_1^1(t) = x(t) - m_1^1(t)$.
- *Step 5*: Examine whether the residual $h_1^1(t)$ satisfies the definition of intrinsic mode function (IMF)[8] according to a stopping criterion.
  - **NO**: Repeat n-times the *Step 2* to *Step 5* until the conditions are met (*sifting* process). Thus: $IMF_1 = h_1^n(t) = h_1^{n-1}(t) - m_1^n(t)$.
  - **YES**: The first IMF is found, $C_1 = h_1^1(t)$.
- *Step 6*: Iterate on the residual, $x(t) - IMF_1 = r_1$.

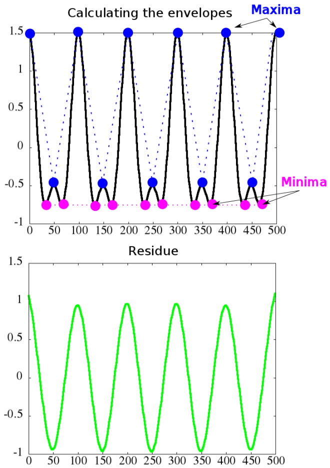


Fig: The sifting process.

[8] IMF-a function having the same number (or differing by one) of zero-crossings and extrema, and symmetric envelopes defined by the local maxima and minima.

Drawback of basic EMD

EMD can introduce *mode mixing* when the local minimum (or maximum) of two signals with different frequency overlap! IMFs consist then of oscillations of dramatically disparate scales, resulting in lack of stability. In other words, during the partial reconstruction, the signal can be filtered out with noise!

Solution

Ensemble EMD- noise-assisted EMD procedure (EEMD); generates multiple noise realisations to keep the *physical uniqueness* of the IMFs. Major steps:

- *Step 1*: Add white noise to the signal.
- *Step 2*: Decompose the data into EMD.
- *Step 3*: Repeat the steps 1 and 2 with different noise, and calculate the ensemble mean.

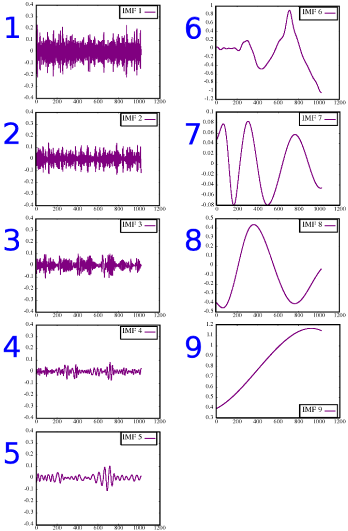


Fig: Decomposition of a noisy signal using EMD.

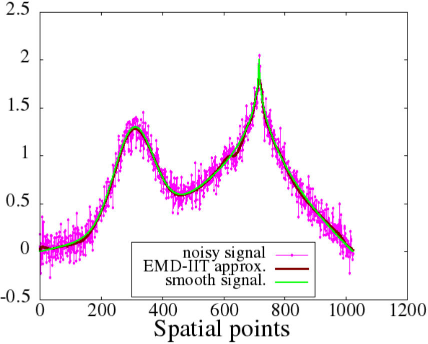


Fig: Result of de-noising Matlab's *cuspamax* signal with EMD-IIT; noisy signal SNR= 18.3052 dB, length=1024; 20 iterations, smooth thresholding. Obtained SNR=32.3534 dB.

Weakness

All EMD-based methods are slow compared to other techniques.
Processing *cuspamax* signal with basic EMD was 9 x slower than wavelet thresholding.

How to choose IMFs that contain signal?

- Visual investigation of IMFs.
- Statistical analysis of modes resulting from signals solely consisting of fractional Gaussian noise and white Gaussian noise.
  - ⇒ Comparing the energy of all IMFs, and discarding those containing only noise.

Can we threshold IMFs like wavelet coefficients?

- EMD interval thresholding (**EMD-IT**).
  $$\tilde{h}^{(i)}\left(z_j^{(i)}\right) = \begin{cases} h^{(i)}\left(z_j^{(i)}\right), & \left|h^{(i)}\left(r_j^{(i)}\right)\right| > T_i \\ 0, & \left|h^{(i)}\left(r_j^{(i)}\right)\right| \leq T_i \end{cases} \text{ for}$$
  $j = 1, 2, \ldots, N_z$, where $h^{(i)}\left(z_j^{(i)}\right)$ indicates samples from the j-th interval between two zero-crossings of the i-th IMF, and $h^{(i)}\left(r_j^{(i)}\right)$ is the single extrema; T_i is a threshold value for i-th IMF.
- Iterative EMD-IT (**EMD-IIT**).
  Altering in a random way the samples of the first IMF. Averaging over different noisy versions of the original signal.

Results of POD+EMD-IIT to particle data

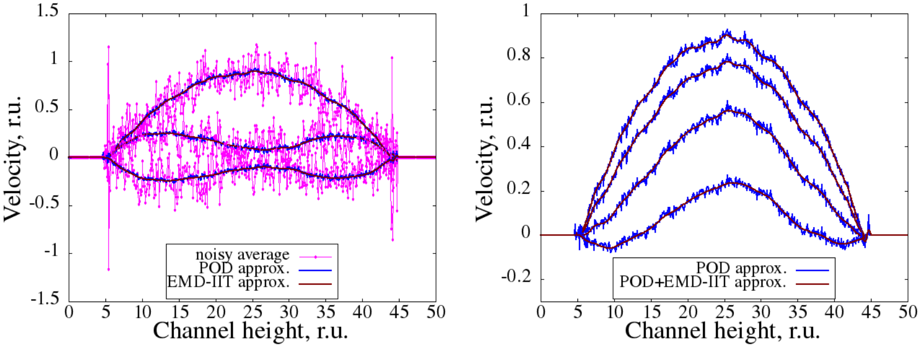


Fig: (*Left* and *Right*)- WPOD compared with POD+EMD-IIT in reconstruction of a velocity profile for oscillating flow of liquid argon from MD simulation; $N_{ts} = 1$, $N_{POD} = 160$ (half a period of oscillation).

- Combination with POD allows for applying EMD-IIT to large matrices, providing improved SNR.
- In studied simulation, the **POD+EMD-IIT** obtained slightly better result than POD+EEMD or EMD. However, the EMD method is the fastest.

- **Wiener filter**- strikes an optimal balance in the bias-variance trade-off, i.e. inverse filtering and noise smoothing.
- It is optimal in terms of mean square error (MSE).
- Wiener filter is an ideal method that requires exact knowledge of the signal and noise statistics prior!

WienerChop (also *WienerShrink*)

- **WienerChop** is based on appropriate adjustment of wavelet coefficients.
- Two wavelet transforms are performed: *first*- to generate noise and signal approximations, and *second*-to filter the wavelet approximation coefficients.

$$F_w(i) = \frac{\widehat{C}_{21}^2(i)}{\widehat{C}_{21}^2(i)+\sigma_n^2},$$

where $i = 1 \ldots N_s$, and N_s is a number of coefficient.

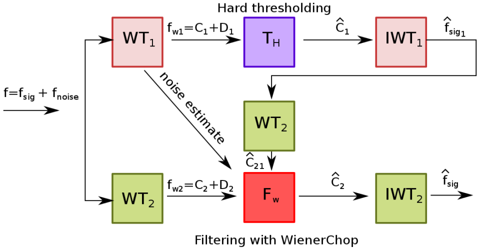


Fig: WienerChop filtering. The first wavelet transform (WT_1) is used to produce the estimate of the signal and noise. The approximations are then used to design an empirical Wiener filter, which is utilised to de-noise the original signal in the WT_2 domain. Inverse transform, IWT_2, is performed to obtain the new data.

WienerChop for de-noising synthetic data

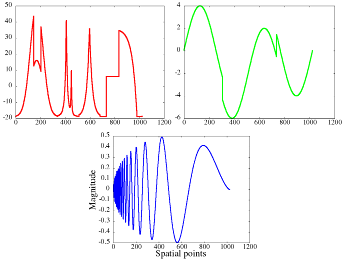


Fig: Matlab's synthetic signals used for analysis: *Piece-Regular* (*Top*), *Heavy-Sine* (*Middle*), and *Doppler*

Cumulative processing time

- WAV≈ **0.024** s
- WienerChop≈ **0.042** s
- SSA≈ **0.017** s
- u/rQRd≈ **0.014** s
- EMD[9]≈ **1.841** s

  ⇒ **WienerChop** recovered the highest SNR, but was 3× slower than u/rRQd.

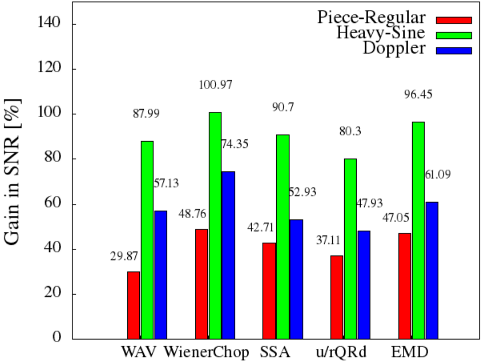


Fig: Gain in SNR obtained with WienerChop and other methods. For wavelet thresholding (WAV) and WT1: db8, 5 levels, hard thresholding; for WT2: db4, 6 levels; for SSA and u/rQRd: $L = 25$; EMD was performed with Iterative interval thresholding (EMD-IIT): 20 iterations and noise variance, $\sigma_n = 0.9$, or just 1 iteration (EMD-IT).

[9] EMD-IIT was used for the first 2 signals, and EMD-IT for the last one.

WienerChop applied to MD data

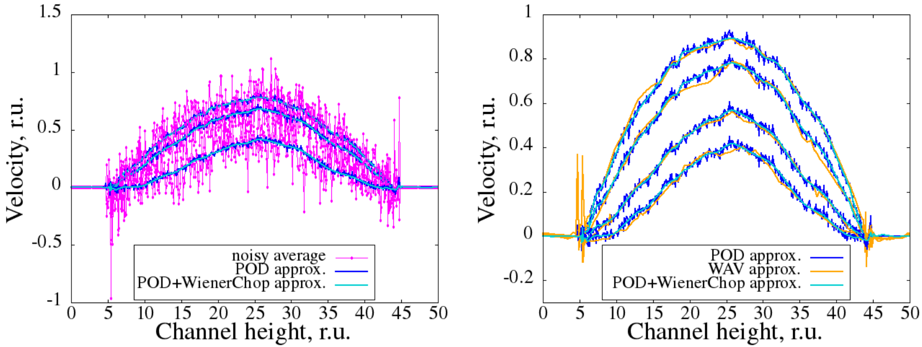


Fig: (*Left* and *Right*)- **WPOD** compared with **POD+WienerChop** and **wavelet thresholding** in reconstruction of a velocity profile for oscillating flow of liquid argon from MD simulation; $N_{ts} = 1$, $N_{POD} = 160$ (half a period of oscillation); for WAV and WT1: db3, 6 levels, soft thresholding; for WT2: db6, 6 levels;

- For large matrices the combination of **POD+WienerChop** is faster than 2-dimensional **WienerChop** and recovers higher SNR.
- In studied simulation, the **POD+WienerChop** provided slight improvement compared to **WAVinPOD**, but it was about 44% slower.

Techniques	Strengths	Weaknesses
POD/WPOD	+ Data-adaptive basis, + No parameters needed, + The most optimal approximation obtained for k.	- Large amount of data needed, - Computationally expensive, - Determination of significant EOFs
SSA/MSSA/2D SSA	+ Data-adaptive basis, + Can be applied to 1D data, + Provides the most optimal solution in L_2 norm.	- Window size defined prior to processing, - Not applicable for large data-sets, - Difficulties in determination of number k.
rQRd/urQRd	+ Very fast in processing large matrices, + Higher flexibility in the choice of EOFs.	- Less optimal solution than SVD.
Wavelet thresholding	+ High SNR, + Fast processing at different resolutions, + Applicable to large matrices and data arrays.	- *A priori* basis, - Conditioned by many parameters.
EMD/EEMD	+ Solely data-dependent, + Simple algorithm, + No parameters needed.	- Can cause mode mixing and costly iterative methods needs to be applied, does not preserve sharp edges.
WienerChop	+ Solely data-dependent, + Retains higher SNR than WT for strongly disturbed data, + Can be applied to 1D signals and large matrices.	- Dependent on the number of parameters, - Pre-determined basis.
WAVinPOD	+ Less dependent on wavelet basis, + Higher SNR than POD or wavelet thresholding alone for additive white noise, + Preserves SVD's dimensionality reduction.	- Choice of the filter and number k, - Slower for the same number of observations.
POD+SSA/MSSA	+ Allows for applying SSA/MSSA to larger data-sets, + Higher SNR than POD alone, + No *a priori* basis needed.	- Computationally more intensive than POD or wavelet thresholding, - Multiple determination of EOFs.
POD+EMD/EMD-IIT	+ Higher SNR than for WPOD for the same number of observations.	- The most expensive combination.
POD+WienerChop	+ Slightly enhanced SNR compared to WAVinPOD for matrices, + The highest SNR for studied signals.	- More expensive than WAVinPOD, - Basis is defined twice.

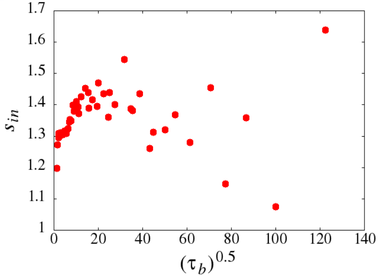
- In particle simulations, due to the small finite time-steps, there is no guarantee that data is independent.
- De-noising methods would be computationally more efficient if the raw data was statistically uncorrelated, i.e. every observation should provide new information.

Statistical Inefficiency

- The sequence of steps is broken up into n_b blocks each of length τ_b, so that $n_b \tau_b =$ total size of the data set.
- The statistical inefficiency s_{in} is defined as:

  $$s_{in} = \lim_{\tau_b \to \infty} \frac{\tau_b \cdot var^2(\langle A \rangle_b)}{var^2(A)},$$

  where $var^2(\langle A \rangle_b)$ and $var^2(A)$ are the block and mean variance, respectively.
- For a time-step Δt, the data should be sampled every time interval $\Delta_s = \Delta t \cdot s_{in}$.



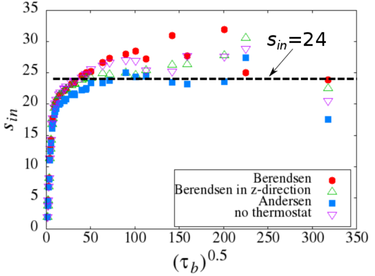


Fig: Calculation of statistical inefficiency s_{in}. Data sampled at: *Top*- every 30th successive measurement; *Bottom*- every time-step for different thermostats.

Summary

Conclusions

- The processing time of obtaining useful particle data can be significantly reduced with de-noising techniques.
- The choice of algorithm = the computational cost v SNR.
- If the nature and number of components of the desired signal is known, the methods utilising *a priori* basis can produce very good de-noising results in terms of time and error in L_2 (or Frobenius) norm.
- Applying noise reduction techniques can largely improve the information transfer in multi-scale simulations.
- Further study is required for processing data corrupted with correlated, $\frac{1}{f}$ noise.

Acknowlegments

The research leading to these results has received funding from the UK's **Engineering and Physical Sciences Research Council** under Grant EP/I011927/1. Results were obtained using the ARCHIE-WeSt High Performance Computer, under EPSRC grant EP/K000586/1.
The authors also express their gratitude to **Špela Ivekovič** for numerous discussions on POD, **Matthew Borg** and **Craig White** for valuable MD and DSMC data.

References

1. Chiron, Lionel, et al. "Efficient denoising algorithms for large experimental datasets and their applications in Fourier transform ion cyclotron resonance mass spectrometry." Proceedings of the National Academy of Sciences 111.4 (2014): 1385-1390.
2. Ghael, Sandeep P., Akbar M. Sayeed, and Richard G. Baraniuk. "Improved wavelet denoising via empirical Wiener filtering." Optical Science, Engineering and Instrumentation'97. International Society for Optics and Photonics, 1997.
3. Golyandina, Nina, and Anatoly Zhigljavsky. Singular Spectrum Analysis for time series. Springer, 2013.
4. Grinberg, Leopold. "Proper orthogonal decomposition of atomistic flow simulations." Journal of Computational Physics 231.16 (2012): 5542-5556.
5. Kopsinis, Yannis, and Steve McLaughlin. "Development of EMD-based denoising methods inspired by wavelet thresholding." Signal Processing, IEEE Transactions on 57.4 (2009): 1351-1362.
6. Mallat, Stephane. A wavelet tour of signal processing: the sparse way. Academic press, 2008.
7. Wu, Zhaohua, and Norden E. Huang. "Ensemble empirical mode decomposition: a noise-assisted data analysis method." Advances in adaptive data analysis 1.01 (2009): 1-41.